Logica (I&E)

najaar 2017

http://liacs.leidenuniv.nl/~vlietrvan1/logica/

Rudy van Vliet
kamer 140 Snellius, tel. 071-527 2876
rvvliet(at)liacs(dot)nl

college 13, maandag 4 december 2017

2. Predicate logic
2.4. Semantics of predicate logic
Semantic tableaux for predicate logic
Soms moet er iets gebeuren voordat er iets gebeurt.

A slide from lecture 12:

Definition 2.14.
Let $\mathcal{F}$ be a set of function symbols and $\mathcal{P}$ a set of predicate symbols, each symbol with a fixed arity.
A model of the pair $(\mathcal{F}, \mathcal{P})$ consists of the following set of data:

1. A non-empty set A, the universe of concrete values;

2. for each nullary symbol $f \in \mathcal{F}$, a concrete element $f^{\mathcal{M}}$ of A;

3. for each $f \in \mathcal{F}$ with arity $n > 0$, a concrete function $f^{\mathcal{M}} : A^n \to A$ from A^n, the set of n-tuples over A, to A;

4. for each $P \in \mathcal{P}$ with arity $n > 0$, a subset $P^{\mathcal{M}} \subseteq A^n$ of n-tuples over A;

5. $=^{\mathcal{M}}$ is equality on A

For all students x: ϕ

There exists a student x: ϕ

Use predicate Student(x)...

A slide from lecture 12:

Definition 2.17.
A look-up table or environment for a universe A of concrete values is a function $l : \textbf{var} \to A$ from the set of variables **var** to A.
For such an l, we denote by $l[x \mapsto a]$ the look-up table which maps x to a and any other variable y to $l(y)$.

A slide from lecture 12:

Definition 2.18.
Given a model $\mathcal{M}$ for a pair $(\mathcal{F}, \mathcal{P})$ and given a look-up table l, we define the satisfaction relation $\mathcal{M} \vDash_l \phi$ for each logical formula ϕ over the pair $(\mathcal{F}, \mathcal{P})$ and look-up table l by structural induction on ϕ.

If $\mathcal{M} \vDash_l \phi$ holds, we say that ϕ computes to T in the model $\mathcal{M}$ with respect to the look-up table l.

A slide from lecture 12:

Definition 2.18. (continued)
P: If ϕ is of the form $P(t_1, t_2, \ldots, t_n)$, then we interpret the terms $t_1, t_2, \ldots, t_n$ in our set A by replacing all variables with their values according to l. In this way we compute concrete values $a_1, a_2, \ldots, a_n$ from A for each of these terms, where we interpret any function symbol $f \in \mathcal{F}$ by $f^{\mathcal{M}}$.
Now $\mathcal{M} \vDash_l P(t_1, t_2, \ldots, t_n)$ holds, iff $(a_1, a_2, \ldots, a_n)$ is in the set $P^{\mathcal{M}}$.

A slide from lecture 12:

Definition 2.18. (continued)

$\forall x$: The relation $\mathcal{M} \vDash_l \forall x \psi$ holds, iff $\mathcal{M} \vDash_{l[x \mapsto a]} \psi$ holds for all $a \in A$.

$\exists x$: The relation $\mathcal{M} \vDash_l \exists x \psi$ holds, iff $\mathcal{M} \vDash_{l[x \mapsto a]} \psi$ holds for some $a \in A$.

A slide from lecture 12:

Definition 2.18. (continued)

$\neg$: The relation $\mathcal{M} \vDash_l \neg\psi$ holds, iff $\mathcal{M} \vDash_l \psi$ does not hold.

$\vee$: The relation $\mathcal{M} \vDash_l \psi_1 \vee \psi_2$ holds, iff $\mathcal{M} \vDash_l \psi_1$ or $\mathcal{M} \vDash_l \psi_2$ holds.

$\wedge$: The relation $\mathcal{M} \vDash_l \psi_1 \wedge \psi_2$ holds, iff $\mathcal{M} \vDash_l \psi_1$ and $\mathcal{M} \vDash_l \psi_2$ holds.

$\rightarrow$: The relation $\mathcal{M} \vDash_l \psi_1 \rightarrow \psi_2$ holds, iff $\mathcal{M} \vDash_l \psi_2$ holds whenever $\mathcal{M} \vDash_l \psi_1$ holds.

2.4.2. Semantic entailment

Definition 2.20.
Let Γ be a (possibly infinite) set of formulas in predicate logic and ψ a formula of predicate logic.

1. Semantic entailment $\Gamma \vDash \psi$, iff for all models $\mathcal{M}$ and look-up tables l, whenever $\mathcal{M} \vDash_l \phi$ holds for all $\phi \in \Gamma$, then $\mathcal{M} \vDash_l \psi$ holds as well.
2. Formula ψ is satisfiable, iff there is some model $\mathcal{M}$ and some look-up table l such that $\mathcal{M} \vDash_l \psi$ holds.
3. Formula ψ is valid, iff $\mathcal{M} \vDash_l \psi$ holds for all models $\mathcal{M}$ and look-up tables l in which we can check ψ, i.e., iff $\vDash \psi$.
4. The set Γ is consistent or satisfiable, iff there is some model $\mathcal{M}$ and and some look-up table l such that $\mathcal{M} \vDash_l \phi$ holds for all $\phi \in \Gamma$.

$\mathcal{M} \models \phi$ vs. $\phi_1, \phi_2, \ldots, \phi_n \models \psi$

Computational . . .

In propositional logic. . .

Example 2.21.

Is

$$\forall x(P(x) \to Q(x)) \models \forall x P(x) \to \forall x Q(x)$$

valid?

Is

$$\forall x P(x) \to \forall x Q(x) \models \forall x(P(x) \to Q(x))$$

valid?

2.4.3. The semantics of equality

Mild requirements on model...

$\phi_1, \phi_2, \ldots, \phi_n \vDash \psi$

Special predicate $=$: $t_1 = t_2$

Semantically, $=^{\mathcal{M}} \;\; = \;\; \ldots$

9. Predikaatlogica: semantische tableaus

[Van Benthem et al]

To find counter example of a *gevolgtrekking*

$$\phi_1, \ldots, \phi_n \;/\; \psi$$

in predicate logic

Predicate $P(x) = Px$ $\qquad R(x, y) = Rxy$

Substitution: $\phi[t/x] = [t/x]\phi$

Definition 2.14.
Let $\mathcal{F}$ be a set of function symbols and $\mathcal{P}$ a set of predicate symbols, each symbol with a fixed arity.
A model of the pair $(\mathcal{F}, \mathcal{P})$ consists of the following set of data:

1. A non-empty set A, the universe of concrete values;
2. for each nullary symbol $f \in \mathcal{F}$, a concrete element $f^{\mathcal{M}}$ of A;
3. for each $f \in \mathcal{F}$ with arity $n > 0$, a concrete function $f^{\mathcal{M}} : A^n \to A$ from A^n, the set of n-tuples over A, to A;
4. for each $P \in \mathcal{P}$ with arity $n > 0$, a subset $P^{\mathcal{M}} \subseteq A^n$ of n-tuples over A;
5. $=^{\mathcal{M}}$ is equality on A

1. = domein D 2–4 = interpretatiefunctie I
look-up table l = bedeling b

Extending semantic tableaux from propositional logic

- reduction rules for $\forall$ and $\exists$
- building up domain D
- building up *interpretatiefunctie* I (and *bedeling* b)

We ignore function symbols (including constants) and free variables.

Voorbeeld 9.1.

$$\forall x(A(x) \to B(x)),\ \forall x(B(x) \to C(x)) \ / \ \forall x(A(x) \to C(x))$$

Valid or not?

Extra reduction rules

Suppose we already have $D = \{d_1, d_2, \ldots, d_k\}$

$\forall_L$:

$$\begin{array}{c} \Phi, \forall x \phi \circ \Psi \\ | \\ \Phi, \phi[d/x] \circ \Psi \end{array}$$

$\forall_R$:

$$\begin{array}{c} \Phi \circ \forall x \phi, \Psi \\ | \\ \Phi \circ \phi[d_{k+1}/x], \Psi \end{array}$$

where d is any existing d_i, and d_{k+1} is new

Voorbeeld 9.2.

$$\forall x(A(x) \to \forall y B(y)) \;/\; \forall x \forall y(A(x) \to B(y))$$

Valid or not?

Voorbeeld 9.3.

Alle kaaimannen zijn reptielen. Geen reptiel kan fluiten.
Dus geen kaaiman kan fluiten.

$$\forall x(K(x) \to R(x)),\ \neg\exists x(R(x) \wedge F(x))\ /\ \neg\exists x(K(x) \wedge F(x))$$

Valid or not?

Extra reduction rules

Suppose we already have $D = \{d_1, d_2, \ldots, d_k\}$

$\forall_{\mathsf{L}}$: $\begin{array}{c} \Phi, \forall x \phi \circ \Psi \\ \mid \\ \Phi, \phi[d/x] \circ \Psi \end{array}$	$\forall_{\mathsf{R}}$: $\begin{array}{c} \Phi \circ \forall x \phi, \Psi \\ \mid \\ \Phi \circ \phi[d_{k+1}/x], \Psi \end{array}$
$\exists_{\mathsf{L}}$: $\begin{array}{c} \Phi, \exists x \phi \circ \Psi \\ \mid \\ \Phi, \phi[d_{k+1}/x] \circ \Psi \end{array}$	$\exists_{\mathsf{R}}$: $\begin{array}{c} \Phi \circ \exists x \phi, \Psi \\ \mid \\ \Phi \circ \phi[d/x], \Psi \end{array}$

where d is any existing d_i, and d_{k+1} is new

Voorbeeld 9.4.

Geen A is B. Geen B is C.
Dus geen A is C.

Geen professor is student. Geen student is gepromoveerd.
Dus geen professor is gepromoveerd.

$$\neg\exists x(A(x) \land B(x)),\ \neg\exists x(B(x) \land C(x))\ /\ \neg\exists x(A(x) \land C(x))$$

Valid or not?

Voorbeeld 9.5.

$$\exists x \forall y R(x,y) \ / \ \forall y \exists x R(x,y)$$

Valid or not?

Voorbeeld 9.6.

$$\forall y \exists x R(x,y) \;/\; \exists x \forall y R(x,y)$$

Valid or not?

Voorbeeld 9.6.

$$\forall y \exists x R(x,y) \ / \ \exists x \forall y R(x,y)$$

Valid or not?

Infinite branch,
which yields counter example with infinite domain.

E.g. $D \stackrel{\text{def}}{=} \mathbb{N}$, $\qquad R^{\mathcal{M}} \stackrel{\text{def}}{=} {'}{>}{'}$

9.3. Een verfijning van de methode

Study this section yourself

9.4. Samenvatting en opmerkingen

Possible situations:

1. Tableau closes (and is finite), hence *gevolgtrekking* is valid

2. There is a non-closing branch

   2.1 finite

   2.2 infinite

   describing counter example

Undecidability

How to decide that we are on an infinite branch?

Adequacy

A *gevolgtrekking* is valid, if and only if there is a closed tableau.